

FIG. 1

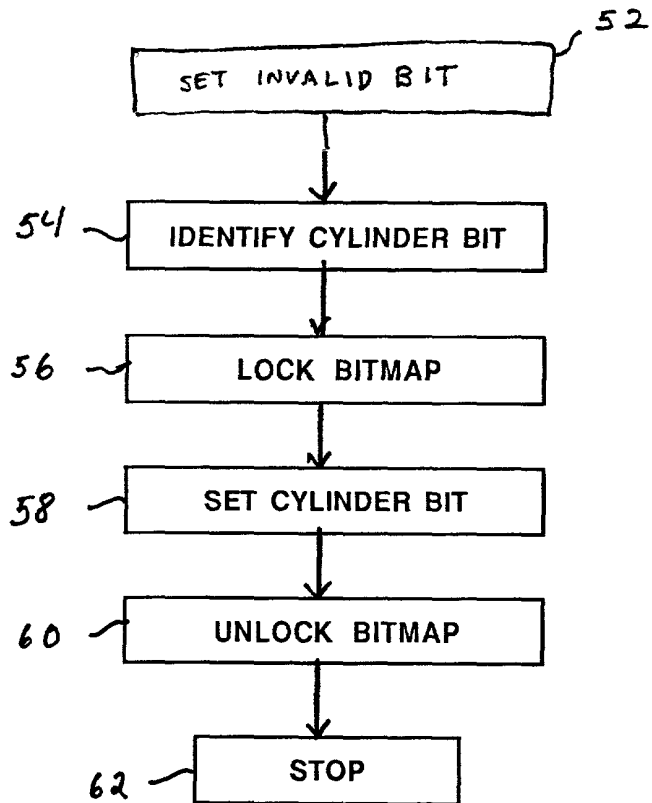


FIG. 2

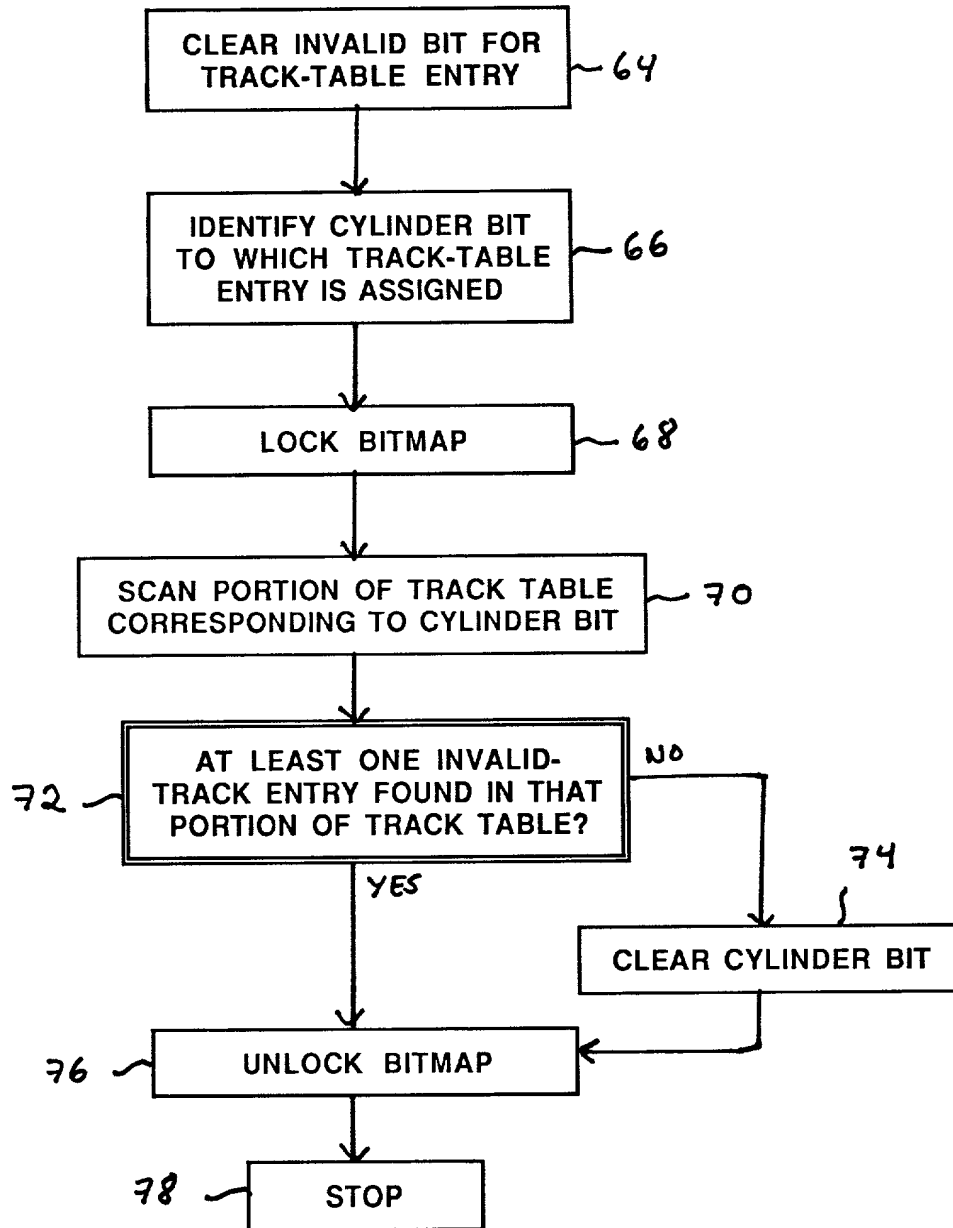


FIG. 3

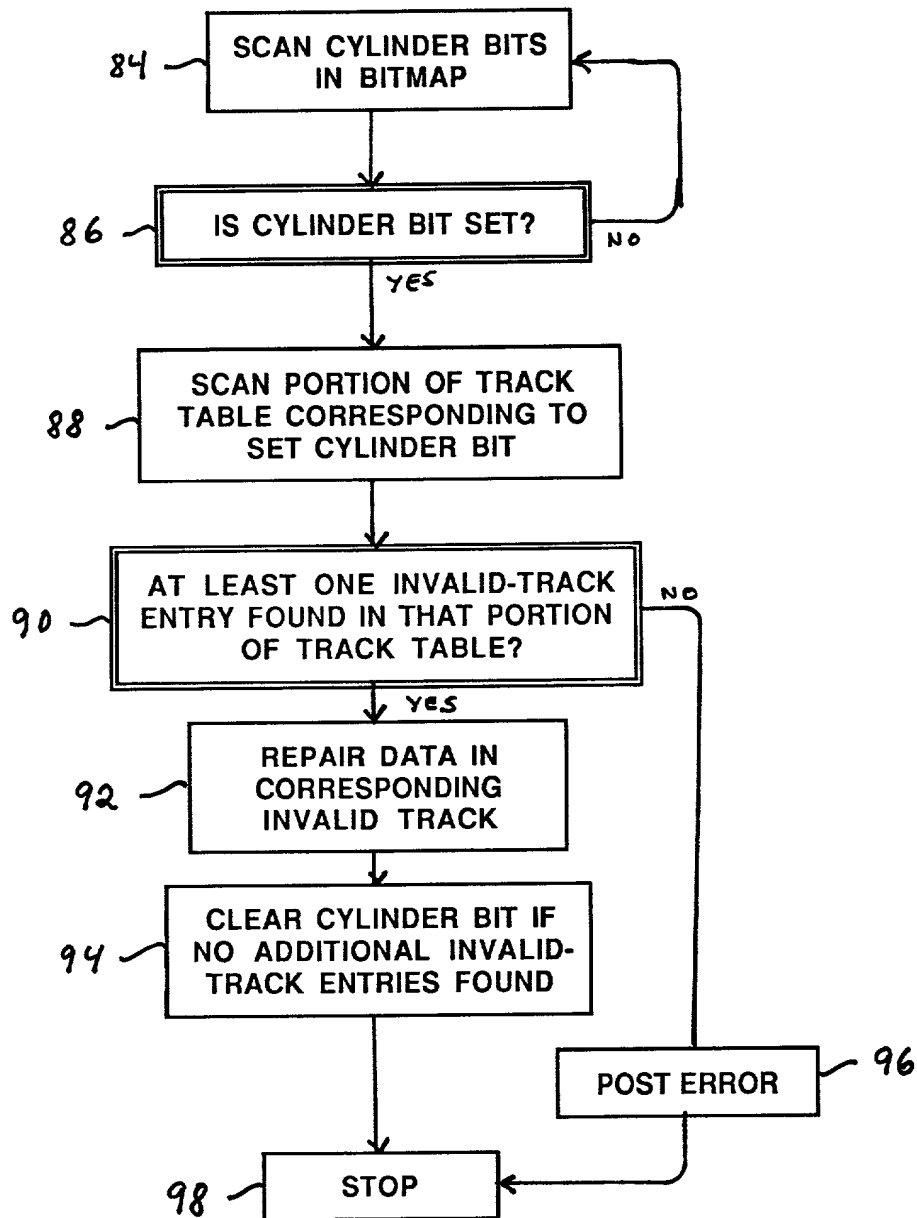


FIG. 4

```
procedure found_invalid_track(T)
100  { O = (T/15)/8
      B = remainder(remainder(T/15)/8)
      X = pointer to bitmap
102  } lock bitmap
      D = byte at offset 0 from X
104  if (bit B of D) = 0 then
      106  set (bit B of D) = 1;
      108  write D at offset 0 from X
      endif
110  unlock bitmap
```

FIG. 5

```
procedure fixed_invalid_track(T)
112 { O = (T/15)/8
      B = remainder(remainder(T/15)/8)
      X = pointer to bitmap
114 } lock bitmap
      D = byte at offset 0 from X
116 { if (bit B of D) = 1 then
117   count = 15
      for each track T1 in cylinder containing track T do
118   if T1 is valid then
119     count=count-1
      endif
    endfor
120 if count=0 then
121   set (bit B of D) = 0
122   write D at offset 0 from X
      else
124   trace error
      endif
123 unlock bitmap
```

FIG. 6

```

126 procedure find_next_invalid_track(T)
127   T = last invalid track
128   X = pointer to bitmap
129   T1 = remainder(T+1)/n_tracks
130   repeat
131     { O=(T1/15)/8
132     { D=byte at offset O from X
133     { if D#0 then
134       for each bit B1 in D do
135         if bit B1=1 then
136           for each track T2 in cylinder containing T1 do
137             if T2 is invalid then
138               return T2
139             endif
140           endfor
141         endif
142       endfor
143       T1 = remainder((T1+15)/n_tracks)
144     else
145       T1=remainder((O*15*8+remainder(T1/15))/n_tracks)
146     endif
147   until T1=T;
148   return -1.

```

FIG. 7